# Deep Learning Approach for Suspicious Activity Detection from Surveillance Video

### A Major Project Report

***Submitted to***



#### JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD, KUKATPALLY, HYDERABAD – 500 085

*in partial fulfilment of the requirements for the award of the degree of*

#### BACHELOR OF TECHNOLOGY

in

#### COMPUTER SCIENCE and ENGINEERING

by

#### BANDI NIHARIKA (19L55A0501)

Under the guidance of

#### Ms. SHAHANAZ BEGUM

Asst. Professor, Department of CSE

### DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**SHADAN WOMEN’S COLLEGE OF ENGINEERING & TECHNOLOGY KHAIRATABAD, HYDERABAD – 500 004.JUNE 2022**

**CERTIFICATE**

This is to certify that the Major Project Report titled **“DEEP LEARNING APPROACH FOR SUSPICIOUS ACTIVITY DETECTION FROM SURVEILLANCE VIDEO”** is being submitted by**” BANDI NIHARIKA (19L55A0501)”**in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** to **Jawaharlal Nehru Technological University Hyderabad, Hyderabad** is a record of bonafide work carried out by her under my guidance and supervision during the academic year 2021 - 2022.

The results presented in this thesis have been verified and are found to be satisfactory. The results embodied in this thesis have not been submitted for the award of any other degree or diploma to this/any other University.

|  |  |
| --- | --- |
| **Ms. SHAHANAZ BEGUM ASST. PROFESSOR**  Department of CSE  Shadan Women’s College of Engineering and Technology, Khairatabad – 500 004. | **Mr. M. K. NIZAMUDDIN HEAD OF THE DEPARTMENT**  Department of CSE  Shadan Women’s College of Engineering and Technology, Khairatabad – 500 004. |

## DECLARATION

I hereby declare that the Major Project Report titled **“DEEP LEARNING APPROACH FOR SUSPICIOUS ACTIVITY DETECTION FROM SURVEILLANCE VIDEO”** is a record of bonafide work done by us in the Department of Computer Science and Engineering, **Shadan Women’s College of Engineering and Technology, Khairatabad,** submitted to the **Jawaharlal Nehru Technological University Hyderabad, Hyderabad** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering**.

The results embodied in this thesis have not been submitted for the award of any other degree or diploma to this/any other University.

**BANDI NIHARIKA (19L55A0501)**

## ACKNOWLEDGEMENT

The satisfaction and euphoria of successful completion of any work could be incomplete without mentioning the persons who made it possible, whose constant guidance and encouragement crown my efforts with success.

I take this opportunity to express my grateful acknowledgement to **The Management,** Shadan Women’s College of Engineering and Technology, Khairatabad for their kind encouragement and granting permission to do this thesis.

I also express my thanks to **Dr. K. Palani, Principal,** Shadan Women’s College of Engineering and Technology, Khairatabad for providing infrastructure and facilities.

I express my sincere gratitude to **Mr. M. K. Nizamuddin, Head of the Department of CSE,** for his extended help, concern and persistent encouragement and support.

I wish to express my heartfelt thanks and sincere acknowledgement to my guide **Ms. Shahanaz Begum, Asst. Professor** for her concern and help to direct me for every move in this thesis.

I acknowledge with grateful thanks to the authors of the references and other literatures referred in this thesis.

I finally thank my parents, friends and relatives who render their help directly or indirectly for the completion of this thesis.

**BANDI NIHARIKA(19L55A0501)**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.No** | **CONTENT** | **PAGE NO** |
|  | **LIST OF FIGURES** | i |
|  | **LIST OF TABLES** | ii |
|  | **ABSTRACT** |  |
| **1** | **INTRODUCTION** | 1 |
| 1.1 | Motivation | 2 |
| 1.2 | Problem definition | 2 |
| 1.3 | scope | 3 |
| **2** | **LITERATURE SURVEY** | 4 |
| 2.1 | Existing system | 10 |
| 2.2 | Proposed system | 11 |
| **3** | **SOFTWARE**  **REQUIRMENT SPECIFICATION** | 12 |
| 3.1 | Problem definition | 12 |
| 3.2 | Solution statement | 12 |

|  |  |  |
| --- | --- | --- |
| 3.3 | Functional requirement | 13 |
| 3.4 | Non-Functional requirement | 14 |
| **4** | **DESIGN** | 15 |
| 4.1 | Data Flow Diagram | 15 |
| 4.2 | UML Diagrams | 16 |
| 4.3 | Database design | 16 |
| **5** | **HARDWARE AND**  **SOFTWARE REQUIREMENTS** | 25 |
| 5.1 | Hardware Requirements | 25 |
| 5.2 | Software Requirements | 26 |
| **6** | **MODULES** | 27 |
| 6.1 | Module Names | 28 |
| 6.2 | Module Explanation | 29 |
| **7** | **IMPLEMENTATION AND**  **CODING** | 32 |
| 7.1 | Implementation | 32 |
| 7.2 | Coding | 32 |
| **8** | **DEPLOYMENT OF THE**  **PROJECT** | 35 |
| 8.1 | Deployment | 35 |
| 8.2 | Steps to Deploy the Website | 35 |
| **9** | **SOFTWARETESTING** | 38 |
| **10** | **OUTPUT SCREENSHOTS** | 41 |
| **11** | **CONCLUSION** | 43 |
| **12** | **REFERENCES** | 44 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **NAME OF THE FIGURE** | **PAGE NO.** |
| 4.1 | Data Dlow Diagram | 15 |
| 4.1.1 | Uscase Diagram | 16 |
| 4.1.2 | Class Diagram | 17 |
| 4.1.3 | Object Diagram | 18 |
| 4.1.4 | State Diagram | 19 |
| 4.1.5 | Sequence Diagram | 20 |
| 4.1.6 | Collaboration Diagram | 21 |
| 4.1.7 | Activity Diagram | 22 |
| 4.1.8 | Component Diagram | 23 |
| 4.1.9 | Deployment Diagram | 24 |
| 6.4 | Architecture Diagram | 31 |

**LIST OF SYMBOLS**



|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **S.NO** | **NOTATION NAME** | **NOTATION** | | | | | | | **DESCRIPTION** |
| 1. | Class |  | | | | | | | Represents a collection of similar entities grouped together. |
| *+ public*  *-private* | | | | *Class Name* | |  |
| *-attribute* | |
| *-attribute* | |
|  | | | | | | |
| 2. | Association |  | | | | | | | Associations represents static relationships between classes. Roles represents the way the two classes see each other. |
|  | Class A | NAME | Class B | |  | |
|  |
|  |  |  |  | |  | |
|  | Class A |  | Class B | |  | |
|  |
|  | | | | | | |
| 3. | Actor |  | | | | | | | It aggregates several classes into a single classes. |
| 4. | Aggregation | Class A Class A  Class B Class B | | | | | | | Interaction between the system and external environment |



|  |  |  |  |
| --- | --- | --- | --- |
| 5. | Relation (uses) | uses | Used for additional process communication. |
| 6. | Relation (extends) | extends | Extends relationship is used when one use case is similar to another use case but does  a bit more. |
| 7. | Communication |  | Communication between various use cases. |
| 8. | State | State | State of the processes. |
| 9. | Initial State |  | Initial state of the object |
| 10. | Final state |  | Final state of the object |
| 11. | Control flow |  | Represents various control flow between the states. |
| 12. | Decision box |  | Represents decision making process from a constraint |

|  |  |  |  |
| --- | --- | --- | --- |
| 13. | Use case | Uses case | Interact ion between the system and external environment. |
| 14. | Component |  | Represents physical modules which are a collection of components. |
| 15. | Node |  | Represents physical modules which are a collection of components. |
| 16. | Data Process/State |  | A circle in DFD represents a state or process which has been triggered due to some event or action. |
| 17. | External entity |  | Represents external entities such as keyboard, sensors,  etc. |
| 18. | Transition |  | Represents communication that occurs between  processes. |
| 19. | Object Lifeline |  | Represents the vertical dimensions that the object  communications. |

|  |  |  |  |
| --- | --- | --- | --- |
| 20. | Message | Message | Represents the message exchanged. |

## ABSTRACT

Human behaviour recognition in the real-world environment finds plenty of applications including intelligent video surveillance, shopping behaviour analysis. Video surveillance has vast application areas especially for indoor outdoor and places. Surveillance is an integral part of security. Today security camera becomes part of life for the safety and security purposes. E- surveillance is one of the main agendas in Digital India, development programme of Indian government. Video surveillance remains as a part of it. Advantages of video surveillance are effective monitoring, less manpower required, cost effective auditing capability, adopting new security trends etc. Currently, the tracking has been performed by human. Since we are dealing with huge amount of video data, this is easy to make people feel tired and the manual intervention will also produce omissions. It greatly affects the efficiency of the system. This has been solved by the automation of video surveillance. Today, manual monitoring of all the events on the CCTV (Closed Circuit Television) camera is impossible. Even if the event had already happened, searching manually the same event in the recorded video wastes a lot of time.

Analysing abnormal events from video is an emerging topic in the domain of automated video surveillance systems.

## CHAPTER 1 INTRODUCTION

Human behaviour recognition in the real-world environment finds plenty of applications including intelligent video surveillance, shopping behaviour analysis. Video surveillance has vast application areas especially for indoor outdoor and places. Surveillance is an integral part of security. Today security camera becomes part of life for the safety and security purposes. E- surveillance is one of the main agendas in Digital India, development programme of Indian government. Video surveillance remains as a part of it. Advantages of video surveillance are effective monitoring, less manpower required, cost effective auditing capability, adopting new security trends etc. Currently, the tracking has been performed by human. Since we are dealing with huge amount of video data, this is easy to make people feel tired and the manual intervention will also produce omissions. It greatly affects the efficiency of the system. This has been solved by the automation of video surveillance. Today, manual monitoring of all the events on the CCTV (Closed Circuit Television) camera is impossible. Even if the event had already happened, searching manually the same event in the recorded video wastes a lot of time. Analysing abnormal events from video is an emerging topic in the domain of automated video surveillance systems.

## MOTIVATION

This main objective of this proposed system will use footages obtained from CCTV camera for monitoring the human behaviour in a campus and gently warn when any suspicious event occurs. The major components in intelligent video monitoring are event detection and human behaviour recognition. Automatic understanding of human behaviour is a challenging task. In a campus, different areas are under video surveillance and various activities are to be monitored. The video footage obtained from campus has been used for testing.

## PROBLEM DEFINITION

Deep Neural Networks is one of the best architecture used to perform difficult learning tasks.

Deep Learning models automatically extract features and builds high level representation of image data. This is more generic because the process of feature extraction is fully automated. From the image pixels, convolutional neural network (CNN) can learn visual patterns directly. In the case of video stream, long short term memory (LSTM) models are capable of learning long term dependencies. LSTM network has the ability to remember things.

The proposed system will use footages obtained from CCTV camera for monitoring the human behavior in a campus and gently warn when any suspicious event occurs. The major components in intelligent video monitoring are event detection and human behavior recognition. Automatic understanding of human behavior is a challenging task. In a campus, different areas are under video surveillance and various activities are to be monitored. The video footage obtained from campus has been used for testing.

## SCOPE

Today manual monitoring of all the events on the CCTV (closed circuit Television) camera is impossible. Even if the event had already happened searching manually the same event in the recorded video wastes a lot of time. Analysing abnormal events from video is an emerging topic in the domain of automated vedio surveillance system.

## CHAPTER 2 LITERATURE SURVEY

**Title:** Inspection of suspicious human activity in the crowdsourced areas captured in surveillance cameras.

**Author:** P.Bhagya Divya, S.Shalini.

**Year:** 2017

**Description:** The ultimate aim is to provide the indoor security using the CCTV camera. The CCTV Camera is a video camera that feeds or streams its image in real time; Webcams are known for their low manufacturing cost and their high flexibility, making them the lowest-cost form of video conversations and inefficient security issues. The system will detect suspicious person i.e. unauthorized entry in a restricted place in a video by using AMD algorithm and will start tracking once the user has specified a suspicious person by his/her on the display. The main purpose of background subtraction is to generate a reliable background model and thus significantly improve the detection of moving objects. Advanced Motion Detection (AMD) achieves complete detection of moving objects. A camera is been connected inside the monitoring room which produces alert messages on the account of any suspicious activity.

**Title:** Suspicious Activity Recognition in Video Surveillance System

**Author:** Prateek Agrawal, Ahmad Salihu Ben Musa.

**Year:** 2014

**Description:** In this research work Suspicious Human Activity Recognition for Video Surveillance System, we detected cheating activities in examination hall. We used SURF (Speed Up Robust Features) to extract interest points, and use SURF method to match and find the corresponding features. We used some algorithms to classify the suspicious activities. We also use Viola Jones object detectors for finding the faces and labelling the activities. We also use tracking algorithms to track detectors in the input video. The proposed techniques use fast detectors and they are robust. In addition to the detectors and tracking algorithms, we used text labelling to avoid false classification, if detectors and tracking algorithms fail to track the faces.

**Title:** Detecting Abnormal Events in University Areas

**Author:** Zahraa Kain, Abir Youness.

**Year:** 2018

**Description:** this paper presents a distinct video surveillance system which took place in the Lebanese International University Saida- Campus, which is considered as a very crowded environment, and reveals if there is an unusual event. Our main target is to apply simple procedures that will be present as a future’s benchmark. The work is split into three major parts, starting by dividing the video frame into zones, then to compute the magnitude of optical flow in each, and finally to analyze these data and classify it, based on a logical threshold, as normal or abnormal events. We implement our results based on Histogram of Magnitudes for each zone (HOM) and the outcome met our expectations.

**Title:** Abnormal Event Detection based on Analysis of Movement Information of Video Sequence

**Author:** Tian Wang, Meina Qiao, Yingjun Deng.

**Year:** 2018

**Description:** Abnormal event detection is a challenging problem in video surveillance which is essential to the early-warning security and protection system. We propose an algorithm to solve this problem efficiently based on an image descriptor which encodes the movement information and the classification method. The new abnormality indicator is derived from the hidden Markov model which learns the histograms of optical flow orientations of the observed video frames.

This indicator measures the similarity between the observed video frame and existing normal frames. The proposed method is evaluated and validated on several video surveillance datasets.

**Title:** Suspicious activity detection using deep learning in secure assisted living IoT environments.

**Author:** G. Vallathan, A. John, Chandrasegar Thirumalai.

**Year:** 2020

**Description:** Children who are left alone in environments such as daycares and crèches require constant monitoring and care to protect them from abuse. In this paper, we propose a novel deep learning-based method for predicting the occurrence of abnormal events using footage gathered from networked surveillance systems and notifying users of those events in an Internet of Things (IoT) environment. Sequences of images are converted to still frames and de-blurred using adaptive motion detection techniques. Then, abnormal activities are predicted using random forest differential evolution with kernel density (RFKD), and any abnormal activities that are detected cause signals to be sent to IoT devices via the MQTT protocol. The proposed work consists of a multi-classifier, deep neural network and kernel density functions. The multi- classifier is used for input classifications from the sequence of frames of videos. The deep neural network is used to learn and train the data and kernel density is used clustering and prediction of data. The novelty of the proposed work is in the dynamic nature of activity prediction. Most of the previous work in this research area concentrated on static activity prediction. The proposed work is able to support both static and dynamic activities of daycare environments. In our

experimental trials, our novel method’s performance is shown to be superior to that of the ReHAR method.

### Existing System

Human behavior detection in video surveillance system is an automated way of intelligently detecting any suspicious activity. Number of efficient algorithms is available for the automatic detection of human behavior in public areas like airports, railway stations, banks, offices, examination halls etc. Video surveillance is the emerging area in the application of Artificial Intelligence, Machine Learning and Deep Learning. Artificial intelligence helps the computer to think like human. In machine learning, important components are learning from the training data and make prediction on future data. Nowadays GPU (Graphics Processing Unit) processors and huge datasets are available, so the concept of deep learning is used.

### Existing System Disadvantages

* + - Performance is less

### Proposed System

A system was developed to monitor students’ behavior in examination using neural network and Gaussian distribution. It consists of three different stages: face detection, suspicious state detection and anomalous detection. The trained model decides whether the student was in suspicious state or not and Gaussian distribution decides whether the student performs any anomalous behavior.

### Proposed System Advantages

* + - Performance is increased

## CHAPTER 3

**SOFTWARE REQUIREMENTS SPECIFICATION**

The proposed system will use footages obtained from CCTV camera for monitoring students’ activities in a campus and send message to the corresponding authority when any suspicious event occurs.

## 3.1 PROBLEM DEFINITION

Deep Neural Networks is one of the best architecture used to perform difficult learning tasks.

Deep Learning models automatically extract features and builds high level representation of image data. This is more generic because the process of feature extraction is fully automated. From the image pixels, convolutional neural network (CNN) can learn visual patterns directly. In the case of video stream, long short term memory (LSTM) models are capable of learning long term dependencies. LSTM network has the ability to remember things.

The proposed system will use footages obtained from CCTV camera for monitoring the human behavior in a campus and gently warn when any suspicious event occurs. The major components in intelligent video monitoring are event detection and human behavior recognition. Automatic understanding of human behavior is a challenging task. In a campus, different areas are under video surveillance and various activities are to be monitored. The video footage obtained from campus has been used for testing.

### 3.2 FUNCTIONAL REQUIREMENTS

A functional requirement defines a function of a software-system or its component. A function is described as a set of inputs, the behavior, Deep learning approach is used to detect suspicious or normal activity in an academic environment, and which sends an alert message to the corresponding authority, in case of predicting a suspicious activity. Monitoring is often performed through consecutive frames which are extracted from the video. The entire framework is divided into two parts. In the first part, the features are computed from video frames and in second part, based on the obtained features classifier predict the class as suspicious or normal.

### NON-FUNCTIONAL REQUIREMENTS

The major non-functional Requirements of the system are as follows

#### Usability

The system is designed with completely automated process hence there is no or less user intervention.

#### Reliability

The system is more reliable because of the qualities that are inherited from the chosen platform java. The code built by using java is more reliable.

#### Performance

This system is developing in the high level languages and using the advanced front-end and back-end technologies it will give response to the end user on client system with in very less time.

#### Supportability

The system is designed to be the cross platform supportable. The system is supported on a wide range of hardware and any software platform, which is having JVM, built into the system.

#### Implementation

The system is implemented in web environment using struts framework. The apache tomcat is used as the web server and windows xp professional is used as the platform. Interface the user interface is based on Struts provides HTML Tag

## CHAPTER 4

### DESIGN

* 1. **Data Flow Diagram Data Flow Diagram:**

Preprocessing

Input data

Feature Extraction

Training dataset

No

Yes

### EXPLANATION:

Prediction/Classification

Testing Data

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

# UML Diagram

### Use Case Diagram

Input

Dataset

Input Video



User

Video Pre-Processing

Splitting Data

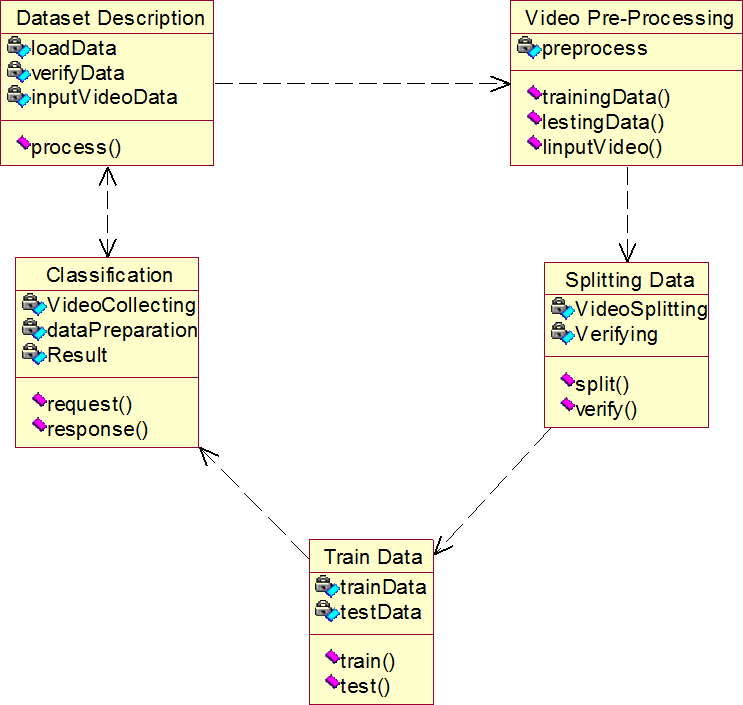
Classification

Result

### EXPLANATION:

The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. The above diagram consists of user as actor. Each will play a certain role to achieve the concept.

### Class Diagram



**EXPLANATION**

In this class diagram represents how the classes with attributes and methods are linked together to perform the verification.

### Object Diagram

Splitting Data : Splitting Data

Video Pre-Processing : Video Pre-Processing

Classification : Classification

Input Video : Input Video

Dataset description : Dataset description

**EXPLANATION:**

In the above digram tells about the flow of objects between the classes. It is a diagram that shows a complete or partial view of the structure of a modeled system. In this object diagram represents how the classes with attributes and methods are linked together to perform the verification with security.

### State Chart Diagram



Input Video

Video Pre-Processing

Splitting Data

Dataset Description

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  | |
|  | CNN (OpenCV) | |  |
|  | |  | |
|  | Classification | | |
|  | |  | |
| Voice Output | | | |

**EXPLANATION:**

State diagram are a loosely defined diagram to show workflows of stepwise activities and actions, with support for choice, iteration and concurrency. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics.

### Sequence Diagram

User

Data Collection

Training

Testing

Load the Data

Collect the data from the user

Send the data to the traing

Perform Preprocessing

Train the data

Extracted feature with images sending to the testing

Give Input

Predict the type using proposed

Display

### EXPLANATION:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

### Collaboration Diagram

****

### EXPLANATION:

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). The concept is more than a decade old although it has been refined as modeling paradigms have evolved.

### Activity Diagram



Dataset Description

Input Video

Video Pre-Processing

Splitting Data

CNN (OpenCV)

Classification

Result

**EXPLANATION:**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

### Component Diagram

Preprocessing

Result

Classificatio n

Data Collection

Data Set Description

Input Video

Splitting Data

Video

Pre-Processing

**EXPLANATION:**

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems. User gives main query and it converted into sub queries and sends through data dissemination to data aggregators. Results are to be showed to user by data aggregators. All boxes are components and arrow indicates dependencies.

### Deployment Diagram:

Processing

Video

Pre-Processing

Input Video

Dataset

Classification

Splitting Data

Data Collection

**EXPLANATION:**

Result

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger deployment and or software systems. They are used to illustrate the structure of arbitrarily complex systems. User gives main query and it converted into sub queries and sends through data dissemination. Results are to be showed to user by data aggregators. All boxes are arrow indicates dependencies.

**CHAPTER 5**

**HARDWARE AND SOFTWARE REQUIREMENTS**

These are the requirements for doing the project. Without using these tools and software’s we can’t do the project. So we have two requirements to do the project. They are

1. Hardware Requirements.
2. Software Requirements.

### HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shouls what the system do and not how it should be implemented.

* + - PROCESSOR : DUAL CORE 2 DUOS.
    - RAM : 4GB DD RAM
    - HARD DISK : 250 GB

### SOFTWARE REQUIREMENTS

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team’s progress throughout the development activity.

* + - OPERATING SYSTEM : WINDOWS 7/8/10
    - PLATFORM : SPYDER3
    - PROGRAMMING LANGUAGE : PYTHON, HTML
    - FRONT END : SPYDER3

## CHAPTER 6

### MODULES

* 1. **MODULE NAMES**

#### Dataset description

* + - **Input Video**

#### Video Pre-Processing

* + - **Splitting Data**
    - **Classification**

### 6.2 Module Explanation

* + - **Dataset Description**

The KTH dataset is a standard dataset which has collection of sequences representing 6 actions and each action class has got 100 sequences. Each sequence has got almost 600 frames and the video is shot at 25 fps [14]. The model is trained on this dataset for normal behaviour (running and walking). CAVIAR dataset, videos taken from campus and YouTube videos are used for training suspicious behaviour (mobile phone using inside the campus, fighting and fainting). Around 7035 frames are extracted from different videos. The whole dataset is manually labelled and separated into 80% for training set and 20% for validation set. The directory structure of dataset is as shown in Fig.2. A combination of KTH, CAVIAR, YouTube videos and videos captured from campus are used in our system

### Input Video

Installation of CCTV camera and monitoring the footage is the initial step in video surveillance system. Various kinds of videos are captured from different cameras, covering the whole area of surveillance. The processing in our implementation is carried out using frames, so the videos are converted to frames

### Video Pre-Processing

A deep learning network is using in our proposed system for suspicious activity detection from video surveillance. By deep learning architectures, the accuracy obtained can be higher and it also works better with large datasets. A detailed design overview is represented in Fig.3. The input videos are taken from existing and created datasets. As part of pre-processing, frames are extracted from the captured videos. Based on the videos, three labelled folders are created and stored the frames in it. The entire video is converted to 7035 frames and the frames are stored as images in jpg format. Each frame is then resized to 224 ×224 to suite 2D CNN architecture and stored them. The testing video is also converted to frames and resized to 224 ×224 and stored in folder. OpenCV library in python is used for video pre- processing

### Splitting Data

The splitting procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data

80% of data used for training 20% of data used for testing

### Classification

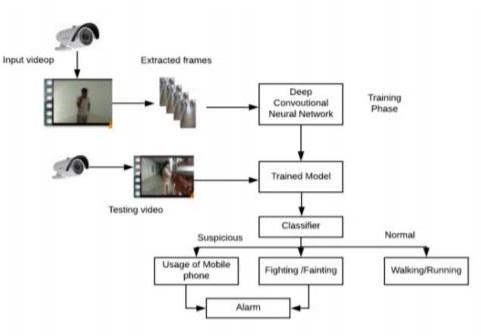
The system classifies the videos as suspicious (students using mobile phone, fighting, fainting) or normal (walking, running). In the case of suspicious behaviour, an SMS (Short Message Service) will be sent to the respective authority.

### TECHNIQUE USED OR ALGORITHM USED

* **VGG-16**

The input videos are taken from CAVIAR dataset, KTH dataset, YouTube videos and videos taken from campus. Around 300 videos of different suspicious and normal behavior videos are collected. As part of pre-processing, frames are extracted from the captured videos. The pre-trained model used in our system is VGG-16 and take its learnings to solve our problem. The last layer of this model is removed based on our requirement and LSTM architecture is used for classification. Our dataset is trained on it. CCTV video footages of different scenarios are taken from our campus for testing and it is converted into frames. The stored frames are given to the trained model and finally the classifier classifies the video into suspicious or normal behavior.

* 1. **System Architecture**



## CHAPTER 7 IMPLEMENTATION & CODING

**6.1 GENERAL**

import cv2

from model import load\_model

import numpy as np

#from scipy.misc import imresize

# import test

from keras.models import load\_model

import argparse

def mean\_squared\_loss(x1,x2):

''' Compute Euclidean Distance Loss between

input frame and the reconstructed frame'''

diff=x1-x2

a,b,c,d,e=diff.shape

n\_samples=a\*b\*c\*d\*e

sq\_diff=diff\*\*2

Sum=sq\_diff.sum()

dist=np.sqrt(Sum)

mean\_dist=dist/n\_samples

return mean\_dist

'''

parser=argparse.ArgumentParser()

parser.add\_argument('modelpath',type=str)

args=parser.parse\_args()

'''

modelpath="path\_to\_model/model.h5"

vc=cv2.VideoCapture("C:/Users/Niharika/Desktop/Work Space/WorkSpace2/WhatsApp Video 2022-06-21 at 12.36.57 PM.mp4") # 0 or "train/1.mp4"

rval=True

print('Loading model')

model=load\_model(modelpath)

print('Model loaded')

threshold=0.065/100

font = cv2.FONT\_HERSHEY\_SIMPLEX

bottomLeftCornerOfText = (8,500)

fontScale = 1

fontColor = (255,0,0)

lineType = 2

while True:

try:

imagedump=[]

for i in range(10):

rval,frame=vc.read()

#frame=imresize(frame,(227,227,3))

frame = cv2.resize(src=frame,dsize=(227,227))

#Convert the Image to Grayscale

gray=0.2989\*frame[:,:,0]+0.5870\*frame[:,:,1]+0.1140\*frame[:,:,2]

gray=(gray-gray.mean())/gray.std()

gray=np.clip(gray,0,1)

imagedump.append(gray)

imagedump=np.array(imagedump)

imagedump.resize(227,227,10)

imagedump=np.expand\_dims(imagedump,axis=0)

imagedump=np.expand\_dims(imagedump,axis=4)

print('Processing data')

output=model.predict(imagedump)

loss=mean\_squared\_loss(imagedump,output)

"""loss = loss\*100"""

print(loss)

frame = cv2.resize(src=frame,

dsize=(1028,1028))

if loss>threshold:

cv2.putText(frame,'Suspicious Detected',

bottomLeftCornerOfText,

font,

fontScale,

fontColor,

lineType)

print('Anomalies Detected')

else:

cv2.putText(frame,'Normal',

bottomLeftCornerOfText,

font,

fontScale,

fontColor,

lineType)

cv2.imshow("output",frame)

key = cv2.waitKey(1)

if key == ord("q"):

cv2.destroyAllWindows()

break

except:

break

cv2.destroyAllWindows()

## CHAPTER 8

### DEVELOPMENT OF THE PROJECT

**Python**

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

### History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

### Importance of Python

* + **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
  + **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
  + **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
  + **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple

text processing to WWW browsers to games.

### Features of Python

* + - **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
      * **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
        + **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
* **A broad standard library** − Python's bulk of the library is very portable and cross- platform compatible on UNIX, Windows, and Macintosh.
* **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
  + **Portable** − Python can run on a wide variety of hardware platforms and has the same

interface on all platforms.

* **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
  + **Databases** − Python provides interfaces to all major commercial databases.
* **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC,

Macintosh, and the X Window system of Unix.

* **Scalable** − Python provides a better structure and support for large programs than shell

scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

* + It supports functional and structured programming methods as well as OOP.
* It can be used as a scripting language or can be compiled to byte-code for building large

applications.

* + It provides very high-level dynamic data types and supports dynamic type checking.
    - IT supports automatic garbage collection.
* It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

### Libraries used in python:

* + numpy - mainly useful for its N-dimensional array objects.
* pandas - Python data analysis library, including structures such as dataframes.
  + matplotlib - 2D plotting library producing publication quality figures.
  + scikit-learn - the machine learning algorithms used for data analysis and data mining

tasks.



Figure : NumPy, Pandas, Matplotlib, Scikit-learn

## CHAPTER 9 SOFTWARE TESTING

### GENERAL

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the

functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

### DEVELOPING METHODOLOGIES

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used. The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

### Types of Tests

#### Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

* + 1. **Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted. Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

#### System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

#### Performance Test

The Performance test ensures that the output be produced within the time limits, and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

#### Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g.,

components in a software system or – one step up – software applications at the company level – interact without error.

* + 1. **Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

#### Acceptance testing for Data Synchronization:

* The Acknowledgements will be received by the Sender Node after the Packets are

received by the Destination Node

* + The Route add operation is done only when there is a Route request in need
* The Status of Nodes information is done automatically in the Cache Updation process

#### 9.2.7 Build the test plan

Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identity the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

## CHAPTER 10





**CHAPTER 11 CONCLUSION**

### 10.1 CONCLUSION

In present world, almost all the people are aware of the importance of CCTV footages, but most of the cases these footages are being used for the investigation purposes after a crime/incident have been happened. The proposed model has the benefit of stopping the crime before it happens. The real time CCTV footages are being tracked and analyzed. The result of the analysis is a command to the respective authority to take an action if in case the result indicates an untoward incident is going to happen. Hence this can be stopped.

Even though the proposed system is limited to academic area, this can also be used to predict more suspicious behaviors at public or private places. The model can be used in any scenario where the training should be given with the suspicious activity suiting for that scenario. The model can be improved by identifying the suspicious individual from the suspicious activity.

## CHAPTER 12

### REFERENCES

1. P.Bhagya Divya, S.Shalini, R.Deepa, Baddeli Sravya Reddy,“Inspection of suspicious human activity in the crowdsourced areas captured in surveillance cameras”,International Research Journal of Engineering and Technology (IRJET), December 2017.
2. Jitendra Musale,Akshata Gavhane, Liyakat Shaikh, Pournima Hagwane, Snehalata Tadge, “Suspicious Movement Detection and Tracking of Human Behavior and Object with Fire Detection using A Closed Circuit TV (CCTV) cameras ”, International Journal for Research in Applied Science & Engineering Technology (IJRASET) Volume 5 Issue XII December 2017.
3. U.M.Kamthe,C.G.Patil “Suspicious Activity Recognition in Video Surveillance System”, Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), 2018.
4. Zahraa Kain, Abir Youness, Ismail El Sayad, Samih Abdul-Nabi, Hussein Kassem, “ Detecting Abnormal Events in University Areas ”, International conference on Computer and Application,2018.
5. Tian Wanga, Meina Qia, Yingjun Deng, Yi Zhouc, Huan Wangd, Qi Lyua, Hichem Snoussie, “Abnormal event detection based on analysis of movement information of video sequence”

,Article-Optik,vol152,January-2018.

1. Elizabeth Scaria, Aby Abahai T and Elizabeth Isaac, “Suspicious Activity Detection in Surveillance Video using Discriminative Deep Belief Netwok”, International Journal of Control Theory and Applications Volume 10, Number 29 -2017.
2. Dinesh Jackson Samuel R,Fenil E, Gunasekaran Manogaran, Vivekananda G.N, Thanjaivadivel T , Jeeva S , Ahilan A, “Real time violence detection framework for football stadium comprising of big data analysis and deep learning through bidirectional LSTM”,The International Journal of Computer and Telecommunications Networking, 2019.
3. Kwang-Eun Ko, Kwee-Bo Sim“Deep convolutional framework for abnormal behaviour detection in a smart surveillance system.”Engineering Applications of Artificial Intelligence ,67 (2018).
4. Yuke Li “A Deep Spatiotemporal Perspective for Understanding Crowd Behavior”, IEEE Transactions on multimedia, Vol. 20, NO. 12, December 2018.
5. Javier Abellan-Abenza, Alberto Garcia-Garcia, Sergiu Oprea, David Ivorra-Piqueres, Jose Garcia-Rodriguez “Classifying Behaviours in Videos with Recurrent Neural Networks”, International Journal of Computer Vision and Image Processing,December 2017.
6. Asma Al Ibrahim, Gibrael Abosamra, Mohamed Dahab “Real-Time Anomalous Behavior Detection of Students in Examination Rooms Using Neural Networks and Gaussian Distribution”, International Journal of Scientific and Engineering Research, October 2018.
7. G. Sreenu and M. A. Saleem Durai “Intelligent video surveillance: a review through deep learning techniques for crowd analysis” , Journal Big Data ,2019.
8. Radha D. and Amudha, J., “Detection of Unauthorized Human Entity in Surveillance Video”, International Journal of Engineering and Technology (IJET), 2013.
9. K. Kavikuil and Amudha, J., “Leveraging deep learning for anomaly detection in video surveillance”, Advances in Intelligent Systems and Computing,2019.
10. Sudarshana Tamuly, C. Jyotsna, Amudha J, “Deep Learning Model for Image Classification”, International Conference on Comp